

EL961412075

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Link Modification System and Method**

Inventors:

Alik Teplitsky

Avihai Lifschitz

Dekel Israeli

Ehud Soen

Oren Trutner

ATTORNEY DOCKET NO. MS1-1871US

## **TECHNICAL FIELD**

The systems and methods described herein relate to modifying links in a document such as a web page.

## **BACKGROUND**

Computer systems are continuing to grow in popularity for business-related uses as well as personal or individual uses. In an organizational setting, computer systems are frequently interconnected with other computer systems via networks, such as local area networks (LANs), wide area networks (WANs) and the Internet. Features such as electronic mail (email), instant messaging, and remote access of organizational servers encourage the use of computer systems coupled to networks. These features allow users to, for example, communicate with other users and retrieve various content, such as documents, product information, and audio or video content.

Entities, such as businesses and other organizations, often utilize an internal (or private) network such as an “intranet”. These internal networks do not typically use the public domain name of the organization. A public domain name is a name that can be resolved over the Internet DNS (Domain Name Service) system. Public domain names often resolve to public IP (Internet Protocol) addresses. Example domain names include microsoft.com, acme.com, or stateuniversity.edu.

Access to internal networks and internal web sites can be made available to users via the Internet (or other data communication network) through a corporate firewall or other mechanism. To access an internal web site (or internal web page) via the Internet, the internal web site is assigned a public URL (Uniform

1 Resource Locator). A user is able to access the internal web site via the Internet  
2 using the public URL. However, problems occur when a user accessing the  
3 internal web site via the Internet activates a link to another internal web site (or  
4 another internal web page). This internal link does not contain the proper  
5 information for an external application (such as a web browser) to access the  
6 appropriate internal web site. For example, an internal web site "http://benefits"  
7 may have a corresponding public URL "http://www.acme.com/benefits". If a user  
8 activates a link embedded in the internal web site to "http://benefits/insurance", an  
9 error message is generated because the link is not a valid public IP address (e.g.,  
10 the link does not contain the public domain name). Further, the internal web site  
11 and the corresponding public web site may contain different data in the path names  
12 or other variations that prevent the internal web site from being accessed via the  
13 Internet.

14 It would be desirable to provide for the proper handling of links to internal  
15 web sites by devices or application programs accessing an internal network via an  
16 external network.

## 17 18 **SUMMARY**

19 The systems and methods described herein modify an internal link in a  
20 document, such as a web page, to allow the document to be accessed by an  
21 external application or device. In a particular embodiment, a process receives a  
22 request for an internal web page from an external browser application. The  
23 process identifies at least one internal link in the internal web page and modifies  
24 the internal link so that the internal link is accessible by the external browser  
25

1 application. The requested web page, including the modified internal link, is then  
2 communicated to the external browser application.

### 3 4 **BRIEF DESCRIPTION OF THE DRAWINGS**

5 Similar reference numbers are used throughout the figures to reference like  
6 components and/or features.

7 Fig. 1 illustrates an example internal web page that contains links to other  
8 web pages on a common intranet.

9 Fig. 2 illustrates an example environment capable of implementing a link  
10 translation process.

11 Fig. 3 illustrates another example environment capable of implementing a  
12 link translation process.

13 Fig. 4 illustrates an example link translation table maintained by a server or  
14 other device in a network.

15 Fig. 5 is a flow diagram illustrating an embodiment of a procedure for  
16 adding string mappings and other translation rules or parameters to a link  
17 translation table.

18 Fig. 6 is a flow diagram illustrating an embodiment of a procedure for  
19 translating links associated with one or more internal web sites.

20 Fig. 7 illustrates a general computer environment, which can be used to  
21 implement the techniques described herein.

## **DETAILED DESCRIPTION**

The systems and methods discussed herein modify links associated with an internal web site or web page such that an external application or device can access the internal web site or web page. Links associated with internal web sites or web pages are translated using information contained in a link translation table. The translated links are provided to the external application or device requesting access to the web site or web page associated with the link. These systems and methods allow external access to internal web pages without having to maintain separate versions of web pages (e.g., an internal version and an external version). Further, the systems and methods discussed herein allow external access to internal web pages without requiring the external application or device to have knowledge of the internal link naming structure or the link translation process.

These systems and methods intercept web pages from internal web sites and replace any links that would not work correctly when accessed via the Internet with properly translated links. For example, link translation can replace the following types of links.

1. Absolute links to web pages on the same internal web site. For example, a link to "http://windows" is replaced with a link to "http://www.microsoft.com/windows".
2. Absolute links that reference a protocol different than the protocol used over the Internet. For example, if the public web site uses HTTPS, but the internal web site uses HTTP, a link to "http://windows" is replaced with a link to "https://www.microsoft.com/windows".

1                   3. Absolute links to other internal web sites that are available through a  
2                   common firewall.

3  
4           Although particular examples discussed herein refer to translating links  
5 contained in internal web pages, alternate embodiments may process any type of  
6 link or other data contained in any type of web page, document, or other collection  
7 of data. Additionally, the examples discussed herein refer to an intranet as an  
8 “internal network” and the Internet as an “external network”. In alternate  
9 embodiments, any network (or portion of a network) may be the “internal  
10 network” and any other network (or portion of a network) may be the “external  
11 network”. Further, one portion of a network may be the “internal network” and  
12 another portion of the same network may be the “external network”.

13           In example embodiments discussed herein, an intranet is also referred to as  
14 an “internal network”. An “intranet” refers to a private (or internal) network of a  
15 company, organization, or other entity. An “intranet web site” or “internal web  
16 site” is a web site installed on an intranet that serves content (e.g., web pages) to  
17 computers and other devices coupled to that intranet. A “public IP address” refers  
18 to an IP address that can be reached via the Internet. A “public domain name”  
19 refers to a domain name that can be resolved over the Internet DNS system.  
20 Public domain names may resolve to public IP addresses. A “public URL” is a  
21 web page URL that contains a public domain name or a public IP address and,  
22 thus, is accessible via the Internet. An “internal IP address” is an IP address that  
23 can be reached via an intranet or other private internal network of an entity. An  
24 “internal domain name” is a domain name that can be resolved over the intranet or  
25 other private network of the entity. Internal domain names may resolve to internal

1 IP addresses. An “internal URL” is a web page URL that contains an internal  
2 domain name and, thus, is accessible via the intranet or other private internal  
3 network of the entity. An “absolute URL” is a URL that contains a domain name,  
4 such as “http://www.microsoft.com/windows/new”. A “relative URL” is a URL  
5 that references a path relative to the page that contains it, such as “windows/new”.  
6 A “root-relative URL” is a URL that references a path relative to the root of the  
7 web site that contains it, such as “/windows/new”.

8 As used herein, “link modification” may also be referred to as “link  
9 translation”, “link conversion”, or “link alteration”. Link modification may also  
10 be referred to as “document modification”. Additionally, the terms “public site”,  
11 “public link” and “public page” may also be referred to as “external site”,  
12 “external link” and “external page”, respectively. The terms “private site”,  
13 “private link” and “private page” may also be referred to as “internal site”,  
14 “internal link” and “internal page”, respectively.

15 Additionally, the term “web page” includes any data (e.g., a document or  
16 collection of information) available via any network or other system. For  
17 example, a web page may be available via an internal network, an external  
18 network, or a combination of an internal network and an external network. As  
19 used herein, a web page is not necessarily publicly accessible via the Internet or  
20 other data communication network.

21 Fig. 1 illustrates an example internal web page 100 that contains links to  
22 other web pages on a common intranet. In this example, web page 100 is  
23 accessible via an internal web site as well as an external web site, such as the  
24 Internet. In one example, the internal URL is “http://benefits” and the  
25

1 corresponding external URL is "http://www.acme.com/benefits". Thus, web page  
2 100 can be accessed internally and externally.

3 Web page 100 contains several internal links 102 associated with other  
4 internal web pages, such as "http://benefits/childcare.html", which is a web page  
5 associated with child care benefits. An application, such as a web browser,  
6 accessing web page 100 via the intranet can successfully activate any of the  
7 internal links 102 and be shown the appropriate web page. But, applications or  
8 devices accessing web page 100 via the Internet (or any other network external to  
9 the intranet) cannot successfully activate the internal links 102 because those  
10 internal links do not contain a full public URL. However, as discussed herein,  
11 internal links 102 are translated, using information contained in a link translation  
12 table, into public URLs that allow external applications or devices to access the  
13 web pages associated with internal links 102. The stored information representing  
14 web page 100 is not modified during the link translation process. Instead, the  
15 internal links 102 are translated as web page 100 is communicated to the  
16 requesting application or device, such that the translated links are communicated  
17 to the requesting application or device instead of the original internal links 102.

18 Fig. 2 illustrates an example environment 200 capable of implementing a  
19 link translation process. A server 202 is coupled to multiple computing devices  
20 204 and 206 via a network 208. Although computing devices 204 and 206 are  
21 shown as computer workstations, computing devices 204 and 206 may be laptop  
22 computers, handheld computers, personal digital assistants (PDAs), cellular  
23 phones, set top boxes, game consoles, or other types of computing devices.  
24 Similarly, server 202 can be any type of computing device, such as the device  
25 discussed below with respect to Fig. 7. Although Fig. 2 illustrates server 202



1 being coupled to two computing devices 204 and 206, server 202 may be coupled  
2 to any number of computing devices, servers, or other devices capable of  
3 communicating with server 202.

4 Network 208 can be any type of data communication network, such as a  
5 local area network (LAN), wide area network (WAN), the Internet, and the like.  
6 In the examples discussed herein, network 208 is referred to as the Internet.  
7 Server 202 is coupled to network 208 via a firewall 210. A broken line 212  
8 identifies the separation of the public Internet from the private intranet (or private  
9 network) to which server 202 is coupled. Firewall 210 is a barrier between the  
10 Internet and the intranet, and is intended to prevent unauthorized access to server  
11 202 and other devices coupled to the intranet. Firewall 210 monitors data flowing  
12 from the Internet to the intranet and blocks certain data or requests to protect the  
13 intranet. Firewall 210 also monitors data flowing from the intranet to the Internet  
14 and translates internal links into external links, as needed. Firewall 210 can be  
15 implemented in hardware, software, or a combination of hardware and software.  
16 Firewall 210 can be a single device or a combination of two or more devices  
17 working together to prevent unauthorized access to the intranet. In alternate  
18 embodiments, firewall 210 is contained in a server, such as server 202.

19 Server 202 includes a translation module 214 that handles various link  
20 translation procedures, such as translating internal links to external links.  
21 Translation module 214 is coupled to firewall 210 as well as a processor 216, a  
22 web page data store 220, a link translation table 222 and a configuration module  
23 224. Translation module 214 may also handle other data conversion and link  
24 translation functions. Processor 216 performs various processing activities  
25 necessary for the operation of server 202. A memory device 218 is coupled to

1 processor 216. Memory device 218 stores various data generated by and/or used  
2 by processor 216 as it performs the various processing activities. Additional  
3 details regarding processor 216 and memory device 218 are discussed below with  
4 reference to Fig. 7.

5 Translation module 214 obtains web page information from web page data  
6 store 220 and translates any local links contained in the web page information  
7 before providing the web page information to firewall 210. Web page data store  
8 220 contains web page information for any number of different web pages. Link  
9 translation table 222 contains information related to translating internal links into  
10 external links that allow external applications or devices to access web pages  
11 associated with the internal links. The information contained in link translation  
12 table 222 is used by translation module 214 in translating links contained in web  
13 pages. Configuration module 224 is used by an administrator or other operator to  
14 define certain link translation rules or other link translation information. The link  
15 translation rules (or link translation information) are stored in link translation table  
16 222 for use by translation module 214. Additional details regarding link  
17 translation table 222, and the information contained therein, are discussed below  
18 with respect to Fig. 4.

19 Translation module 214 and configuration module 224 may be  
20 implemented in software, hardware, or a combination of software and hardware.  
21 Additionally, a particular intranet may include any number of servers 202 and any  
22 number of firewalls 210 coupled to one another. For example, an intranet may  
23 include multiple servers coupled to a single firewall, such that the firewall  
24 processes all data flowing between the Internet and any of the servers in the  
25 intranet. In this embodiment, a particular server (or group of servers) may be

1 responsible for performing link translation functions. Thus, it is not necessary for  
2 each server to contain a translation module 214 and a link translation table 222.

3 Fig. 3 illustrates another example environment 300 capable of  
4 implementing a link translation process. In this embodiment, the translation  
5 module, link translation table and configuration module are contained in a firewall  
6 rather than in a server (as shown in Fig. 2). A firewall 310 is coupled to multiple  
7 servers 302 and coupled to multiple computing devices 304 and 306 via a network  
8 308. Servers 302 and computing devices 304 and 306 may be any type of  
9 computing device, such as the device discussed below with respect to Fig. 7.  
10 Although Fig. 3 illustrates two servers 302 coupled to firewall 310, any number of  
11 servers 302 may be coupled to firewall 310. Further, alternate embodiments of  
12 environment 300 may include any number of firewalls 310 coupled to any number  
13 of servers 302 and any number of networks 308.

14 Network 308 can be any type of data communication network, such as a  
15 LAN, WAN, the Internet, and the like. In the examples discussed below, network  
16 308 is referred to as the Internet. Servers 302 are coupled to network 308 via  
17 firewall 310. A broken line 312 identifies the separation of the public Internet  
18 from the private intranet (or private network) to which servers 302 are coupled.  
19 Firewall 310 is a barrier between the Internet and the intranet, and is intended to  
20 prevent unauthorized access to servers 302 and other devices coupled to the  
21 intranet. For example, firewall 310 monitors data flowing from the Internet to the  
22 intranet and blocks certain data or requests to protect the intranet. Firewall 310  
23 also monitors data flowing from the intranet to the Internet and translates internal  
24 links into external links, as needed. Firewall 310 can be implemented in hardware,  
25 software, or a combination of hardware and software. Firewall 310 can be a single

1 device or a combination of two or more devices working together to prevent  
2 unauthorized access to the intranet.

3 Firewall 310 includes a translation module 314 that handles various link  
4 translation procedures, such as translating internal links to external links.  
5 Translation module 314 may also handle other data conversion and link translation  
6 functions. Translation module 314 is coupled to a link translation table 322 and a  
7 configuration module 324. Link translation table 322 contains information related  
8 to translating internal links into external links that allow external applications or  
9 devices to access web pages associated with the internal links. The information  
10 contained in link translation table 322 is used by translation module 314 to  
11 translate links contained in web pages. Configuration module 324 is used by an  
12 administrator or other operator to define certain link translation rules or other link  
13 translation information. The link translation rules (or link translation information)  
14 are stored in link translation table 322 for use by translation module 314.  
15 Additional details regarding link translation table 322, and the information  
16 contained therein, are discussed below with respect to Fig. 4.

17 Each server 302 includes a processor 316, a memory device 318 and a web  
18 page data store 320. Alternate embodiments of server 302 may omit the web page  
19 data store 320 (e.g., for servers that are not responsible for processing web pages).  
20 Processor 316 performs various processing activities necessary for the operation of  
21 server 302. Memory device 318 stores various data generated by and/or used by  
22 processor 316 as it performs the various processing activities. Additional details  
23 regarding processor 316 and memory device 318 are discussed below with  
24 reference to Fig. 7.

1 In operation, translation module 314 obtains web page information from  
2 one or more web page data stores 320 and translates any local links contained in  
3 the web page information before providing the web page information to a  
4 requesting application program or computing device. Translation module 314 and  
5 configuration module 324 may be implemented in software, hardware, or a  
6 combination of software and hardware.

7 Fig. 4 illustrates an example link translation table 400 maintained by a  
8 server or other device in a network. In one embodiment, link translation table 400  
9 is stored in a server, such as server 202 in Fig. 2. Alternatively, link translation  
10 table 400 may be stored in another device or component, such as firewall 310 in  
11 Fig. 3. In one embodiment, link translation table 400 is stored in the same device  
12 or component as the translation module that accesses link translation table 400.  
13 However, in alternate embodiments, link translation table 400 and the translation  
14 module that accesses the link translation table are located in different devices or  
15 components.

16 A first column 402 of link translation table 400 identifies strings of  
17 characters (or other identifiers) associated with an entire internal link or a portion  
18 of an internal link. A second column 404 of link translation table 400 identifies  
19 additional strings of characters (or other identifiers) associated with an entire  
20 external link or a portion of an external link. Strings in the same row of link  
21 translation table 400 represent translations that are applied to internal links to  
22 make those internal links accessible externally (e.g., via the Internet). For  
23 example, a row 406 indicates that an internal link containing a reference to port 79  
24 should be translated to an external link containing a reference to port 80. A row  
25 408 indicates that an internal link containing a string "HR\_Dept" should be

1 translated to “www.acme.com/HR”. Thus, an internal link “http://HR\_Dept”  
2 would be translated to an external link “http://www.acme.com/HR”.

3 A row 410 in link translation table 400 indicates that internal links  
4 containing a reference to “server1” are translated to external links containing a  
5 corresponding reference to “server2”. A row 412 indicates a change in protocol  
6 indicators – any internal links containing a string “http://” are translated to external  
7 links containing a replacement string “https://”. The strings “http://” and “https://”  
8 are also referred to as “protocol designators” or “protocol indicators”. Rows 414  
9 and 416 illustrate additional examples of particular strings that are translated  
10 between internal links and external links.

11 A particular link translation table 400 may include any number of entries  
12 (e.g., rows). These entries may be defined by an administrator or other user, or  
13 may be defined by a link translation module, as discussed below. In the  
14 embodiment of Fig. 4, user-defined entries and entries defined by the link  
15 translation module are stored in the same link translation table 400. In alternate  
16 embodiments, user-defined entries may be stored in one link translation table and  
17 entries defined by the link translation module are stored in a second link  
18 translation table. In particular embodiments, other components, devices, or  
19 modules may generate one or more entries contained in link translation table 400.

20 Fig. 4 illustrates one embodiment of a link translation table structure. In  
21 alternate embodiments, any type of data structure or data storage arrangement may  
22 be used to store various information used in translating one or more links. For  
23 example, arrays, hash tables, or binary trees can store information used in  
24 translating links.  
25

1           Fig. 5 is a flow diagram illustrating an embodiment of a procedure 500 for  
2 adding string mappings and other translation rules or parameters to a link  
3 translation table. These mappings may be added by a developer, a network  
4 administrator, or other user. Initially, a user creates or identifies an internal web  
5 page or web site (block 502). The internal web page or web site is typically  
6 accessible via an intranet or other internal network. Additionally, the internal web  
7 page or web site may be accessed externally via the Internet or other public  
8 network.

9           Procedure 500 continues by identifying any internal links in the web page  
10 or web site (block 504). This identification of internal links can be performed, for  
11 example, by a user or by a component, such as a translation module. Next, the  
12 procedure creates string mappings that translate the internal links to corresponding  
13 external links (block 506). The string mappings may include an entire link or a  
14 portion of a link. Example string mappings include protocol translations, server  
15 name translations, path name translations, and the like. Additional mappings  
16 include inserting a server name into a link, inserting information into a link and  
17 deleting information from a link. Procedure 500 continues by creating other link  
18 translation rules or link translation parameters, if necessary (block 508). These  
19 link translation rules/parameters include, for example, mapping a port number  
20 associated with an internal link to a different port number in the external link. The  
21 creation of string mappings and other link translation rules/parameters can be  
22 performed, for example, by a user or by a component, such as a translation  
23 module. Finally, the string mappings and other link translation rules/parameters  
24 are added to a link translation table (block 510) for use in translating internal links  
25 to external links.

1       The procedure of Fig. 5 can be performed each time a new web page or  
2 web site is created or added to a server, such as a web server. Additionally, the  
3 procedure of Fig. 5 can be performed when initializing a new server, when  
4 establishing a new network (e.g., an internal network or an external network), or  
5 when adding functionality that permits external application programs or devices to  
6 access existing internal web pages or web sites.

7       In a particular embodiment, a configuration application or configuration  
8 module assists an administrator with creating and storing link translation  
9 information in the link translation table. For example, the configuration  
10 application may request one or more of the following: the computer name or  
11 internal IP address that is being published (e.g., made available externally via the  
12 Internet), a public domain name associated with the internal IP address, a  
13 particular folder associated with the public domain name, IP addresses and ports  
14 on which the translation module “listens” for incoming web requests, absolute  
15 links (or absolute URLs) to replace, and partial links to replace (e.g., strings  
16 representing a portion of a link). The various link translation mappings, rules and  
17 parameters may be applied to selected users requesting internal web pages. For  
18 example, link translation may be performed for a request from a particular user,  
19 but such link translation may be denied for a similar request from a different user.  
20 After an administrator provides information to the configuration application,  
21 appropriate entries are created and stored in the link translation table.

22       Fig. 6 is a flow diagram illustrating an embodiment of a procedure 600 for  
23 translating links associated with one or more internal web pages or web sites.  
24 Initially, a request is received from an application program (such as a browser) or  
25 other device to access a public web site (block 602). The received request may



1 identify a web site (e.g., a home page for the web site) or a particular web page  
2 associated with a web site. Procedure 600 then performs two groups of operations  
3 simultaneously. In one group of operations, the procedure accesses the requested  
4 web site (block 604) and retrieves an internal version of a web page associated  
5 with the requested web site (block 606). In a second group of operations, the  
6 procedure identifies information contained in the received request (block 608).  
7 This information includes, for example, links contained in the request and tags that  
8 identify one or more links. Additionally, information may be retrieved from  
9 header information associated with the received request.

10 An example of information that can be retrieved from a request is provided  
11 below.

12  
13 GET/HTTP/1.1  
14 Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,  
15 application/vnd.ms-excel, application/vnd.ms-powerpoint,  
16 application/msword, application/x-shockwave-flash, \*/\*  
17 Accept-Language: en-us  
18 Accept-Encoding: gzip, deflate  
19 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET  
20 CLR 1.1.4322)  
21 Host: www.abc123.com  
22 Connection: Keep-Alive  
23  
24  
25

19 In the above example, the host is identified as "www.abc123.com". In this  
20 example, a translation module (or other link translation device or algorithm) is  
21 configured to redirect requests for "www.abc123.com" to an internal server named  
22 "abcweb". Thus, when the above information is retrieved from the request, an  
23 entry is added to the link translation table indicating a translation from "abcweb"  
24 to "www.abc123.com".  
25

1 Referring again to Fig. 6, procedure 600 continues by generating link  
2 translation data based on the identified information (block 610). Next, the  
3 procedure stores the link translation data in the link translation table (block 612).  
4 When the two groups of operations are completed, procedure 600 searches the  
5 internal web page for string matches (or other link translation rule/parameter  
6 matches) and replaces those strings with alternate strings from the link translation  
7 table (block 614). The procedure then provides the web page data (including any  
8 translated links) to the requesting browser (block 616). At this point, any links  
9 displayed in the web page on the browser have been translated into public links  
10 (i.e., external links) that can be accessed by the browser via the Internet. Finally,  
11 procedure 600 deletes the link translation data associated with the received request  
12 from the link translation table (block 618). Any user-defined data contained in the  
13 link translation table is maintained for processing future internal links.  
14 Additionally, any link translation data associated with other requests from  
15 browsers is maintained in the link translation table until processing of the other  
16 requests is completed.

17 When searching internal web pages for links that match entries in the link  
18 translation table, the various entries in the link translation table are processed in a  
19 particular order. In one embodiment, user-defined string mappings or other link  
20 translation rules/parameters are searched and applied prior to string mappings or  
21 other link translation rules/parameters generated by the translation module. Thus,  
22 user-defined mappings, rules and parameters have a higher priority than translation  
23 module-generated mappings, rules and parameters. In another embodiment, the  
24 string mappings and other rules/parameters contained in the link translation table  
25 are searched and applied in the order listed in the link translation table. In this

1 embodiment, mappings, rules and parameters at the “top” of the link translation  
2 table (e.g., the top rows shown in Fig. 4) have a higher priority than mappings,  
3 rules and parameters located “lower” in the link translation table. In other  
4 embodiments, alternate procedures are used to determine an order for searching  
5 and applying the various entries in the link translation table.

6 Certain types of data requested by a browser may not be searched for links  
7 requiring translation. For example, pictures, video data and audio data are not  
8 generally searched for internal links needing translation. Additionally, certain  
9 types of response codes (or return codes) are not translated. For example, a  
10 response having a return code of 206 is not translated. This return code implies a  
11 successful response for a request for only a portion of a document (e.g., range  
12 request). In this situation, the translation module or other component does not  
13 have enough information to know whether there may have been links (or portions  
14 of links) that were cut-off at the beginning or end of the portion of the document  
15 contained in the response. In another embodiment, the translation module requests  
16 that the web server send back a larger portion of the document (e.g., a larger  
17 range) to better determine whether any links were cut-off. An alternative  
18 embodiment may go ahead and perform the translation, even if the portion of the  
19 document contained in the response may include one or more links that were cut-  
20 off.

21 Additional return codes that may cause a particular response not to be  
22 translated are: return code 204 (no content), return code 304 (not modified), and  
23 return codes greater than 100 and less than 200 (informational return codes).

24 In particular embodiments, an administrator or other user can determine the  
25 type of data that is searched for internal links. For example, an administrator can

1 determine whether to search the following types of data for internal links:  
2 applications, application data files, audio data, compressed files, documents (e.g.,  
3 word processing documents), HTML documents, images (e.g., pictures), macro  
4 documents, text, video data, and VRML (Virtual Reality Modeling Language).

5 Particular examples discussed herein refer to translating internal network  
6 links to external links accessible via the Internet or other public network.  
7 However, the systems and methods discussed herein can also be used in other  
8 environments, such as scanning XML content and translating other types of  
9 network traffic using any type of naming format.

10 In one embodiment, a translation tool (e.g., an application program) or  
11 other component may be implemented as an ISAPI (Internet Server API) filter.  
12 The translation tool monitors outgoing data from, for example, a publishing proxy  
13 to a client, and modifies links contained in the data. The translation tool can be  
14 configured to translate links in certain types of documents. For example, links in  
15 text and html documents may be translated, whereas links in application programs  
16 or Java scripts are not translated. Additionally, if a response does not have a  
17 content-type header (or if the content type cannot be translated), the translation  
18 tool tries to infer the content type based on the file extension or other information  
19 in the response. If no type is determined or inferred, the response may be sent  
20 without any translation.

21 If the response has a transfer-encoding header with a value equal to  
22 "chunked", the response is sent in chunked transfer-encoding without  
23 accumulating. Additionally, if the response has a content-length header and the  
24 client utilizes HTTP 1.1 or higher, the response is sent in chunked transfer-  
25 encoding without accumulating. But, if the response has a content-length header

1 and the client does not utilize HTTP 1.1 or higher, the response is accumulated,  
2 translated, and sent using the appropriate content length. In a particular  
3 embodiment, the response is accumulated if the size does not exceed a threshold.  
4 If the size does exceed the threshold, the response is not accumulated. When the  
5 size exceeds the threshold, the content-length header is removed, the response is  
6 sent, and the connection is closed after the entire response is sent. If the response  
7 does not have a content-length header, the response is translated “on the fly”  
8 without accumulating.

9 Fig. 7 illustrates a general computer environment 700, which can be used to  
10 implement the techniques described herein. The computer environment 700 is  
11 only one example of a computing environment and is not intended to suggest any  
12 limitation as to the scope of use or functionality of the computer and network  
13 architectures. Neither should the computer environment 700 be interpreted as  
14 having any dependency or requirement relating to any one or combination of  
15 components illustrated in the example computer environment 700.

16 Computer environment 700 includes a general-purpose computing device in  
17 the form of a computer 702. One or more media player applications can be  
18 executed by computer 702. The components of computer 702 can include, but are  
19 not limited to, one or more processors or processing units 704 (optionally  
20 including a cryptographic processor or co-processor), a system memory 706, and a  
21 system bus 708 that couples various system components including the processor  
22 704 to the system memory 706.

23 The system bus 708 represents one or more of any of several types of bus  
24 structures, including a memory bus or memory controller, a point-to-point  
25 connection, a switching fabric, a peripheral bus, an accelerated graphics port, and

1 a processor or local bus using any of a variety of bus architectures. By way of  
2 example, such architectures can include an Industry Standard Architecture (ISA)  
3 bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a  
4 Video Electronics Standards Association (VESA) local bus, and a Peripheral  
5 Component Interconnects (PCI) bus also known as a Mezzanine bus.

6 Computer 702 typically includes a variety of computer readable media.  
7 Such media can be any available media that is accessible by computer 702 and  
8 includes both volatile and non-volatile media, removable and non-removable  
9 media.

10 The system memory 706 includes computer readable media in the form of  
11 volatile memory, such as random access memory (RAM) 710, and/or non-volatile  
12 memory, such as read only memory (ROM) 712. A basic input/output system  
13 (BIOS) 714, containing the basic routines that help to transfer information  
14 between elements within computer 702, such as during start-up, is stored in ROM  
15 712. RAM 710 typically contains data and/or program modules that are  
16 immediately accessible to and/or presently operated on by the processing unit 704.

17 Computer 702 may also include other removable/non-removable,  
18 volatile/non-volatile computer storage media. By way of example, Fig. 7  
19 illustrates a hard disk drive 716 for reading from and writing to a non-removable,  
20 non-volatile magnetic media (not shown), a magnetic disk drive 718 for reading  
21 from and writing to a removable, non-volatile magnetic disk 720 (e.g., a “floppy  
22 disk”), and an optical disk drive 722 for reading from and/or writing to a  
23 removable, non-volatile optical disk 724 such as a CD-ROM, DVD-ROM, or other  
24 optical media. The hard disk drive 716, magnetic disk drive 718, and optical disk  
25 drive 722 are each connected to the system bus 708 by one or more data media

1 interfaces 725. Alternatively, the hard disk drive 716, magnetic disk drive 718,  
2 and optical disk drive 722 can be connected to the system bus 708 by one or more  
3 interfaces (not shown).

4 The disk drives and their associated computer-readable media provide non-  
5 volatile storage of computer readable instructions, data structures, program  
6 modules, and other data for computer 702. Although the example illustrates a hard  
7 disk 716, a removable magnetic disk 720, and a removable optical disk 724, it is to  
8 be appreciated that other types of computer readable media which can store data  
9 that is accessible by a computer, such as magnetic cassettes or other magnetic  
10 storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or  
11 other optical storage, random access memories (RAM), read only memories  
12 (ROM), electrically erasable programmable read-only memory (EEPROM), and  
13 the like, can also be utilized to implement the example computing system and  
14 environment.

15 Any number of program modules can be stored on the hard disk 716,  
16 magnetic disk 720, optical disk 724, ROM 712, and/or RAM 710, including by  
17 way of example, an operating system 726, one or more application programs 728,  
18 other program modules 730, and program data 732. Each of such operating  
19 system 726, one or more application programs 728, other program modules 730,  
20 and program data 732 (or some combination thereof) may implement all or part of  
21 the resident components that support the distributed file system.

22 A user can enter commands and information into computer 702 via input  
23 devices such as a keyboard 734 and a pointing device 736 (e.g., a “mouse”).  
24 Other input devices 738 (not shown specifically) may include a microphone,  
25 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and

1 other input devices are connected to the processing unit 704 via input/output  
2 interfaces 740 that are coupled to the system bus 708, but may be connected by  
3 other interface and bus structures, such as a parallel port, game port, or a universal  
4 serial bus (USB).

5 A monitor 742 or other type of display device can also be connected to the  
6 system bus 708 via an interface, such as a video adapter 744. In addition to the  
7 monitor 742, other output peripheral devices can include components such as  
8 speakers (not shown) and a printer 746 which can be connected to computer 702  
9 via the input/output interfaces 740.

10 Computer 702 can operate in a networked environment using logical  
11 connections to one or more remote computers, such as a remote computing device  
12 748. By way of example, the remote computing device 748 can be a personal  
13 computer, portable computer, a server, a router, a network computer, a peer device  
14 or other common network node, game console, and the like. The remote  
15 computing device 748 is illustrated as a portable computer that can include many  
16 or all of the elements and features described herein relative to computer 702.

17 Logical connections between computer 702 and the remote computer 748  
18 are depicted as a local area network (LAN) 750 and a general wide area network  
19 (WAN) 752. Such networking environments are commonplace in offices,  
20 enterprise-wide computer networks, intranets, and the Internet.

21 When implemented in a LAN networking environment, the computer 702 is  
22 connected to a local network 750 via a network interface or adapter 754. When  
23 implemented in a WAN networking environment, the computer 702 typically  
24 includes a modem 756 or other means for establishing communications over the  
25 wide network 752. The modem 756, which can be internal or external to computer



1 702, can be connected to the system bus 708 via the input/output interfaces 740 or  
2 other appropriate mechanisms. It is to be appreciated that the illustrated network  
3 connections are exemplary and that other means of establishing communication  
4 link(s) between the computers 702 and 748 can be employed.

5 In a networked environment, such as that illustrated with computing  
6 environment 700, program modules depicted relative to the computer 702, or  
7 portions thereof, may be stored in a remote memory storage device. By way of  
8 example, remote application programs 758 reside on a memory device of remote  
9 computer 748. For purposes of illustration, application programs and other  
10 executable program components such as the operating system are illustrated herein  
11 as discrete blocks, although it is recognized that such programs and components  
12 reside at various times in different storage components of the computing device  
13 702, and are executed by the data processor(s) of the computer.

14 Various modules and techniques may be described herein in the general  
15 context of computer-executable instructions, such as program modules, executed  
16 by one or more computers or other devices. Generally, program modules include  
17 routines, programs, objects, components, data structures, etc. that perform  
18 particular tasks or implement particular abstract data types. Typically, the  
19 functionality of the program modules may be combined or distributed as desired in  
20 various embodiments.

21 An implementation of these modules and techniques may be stored on or  
22 transmitted across some form of computer readable media. Computer readable  
23 media can be any available media that can be accessed by a computer. By way of  
24 example, and not limitation, computer readable media may comprise "computer  
25 storage media" and "communications media."

1       “Computer storage media” includes volatile and non-volatile, removable  
2 and non-removable media implemented in any method or technology for storage  
3 of information such as computer readable instructions, data structures, program  
4 modules, or other data. Computer storage media includes, but is not limited to,  
5 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,  
6 digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic  
7 tape, magnetic disk storage or other magnetic storage devices, or any other  
8 medium which can be used to store the desired information and which can be  
9 accessed by a computer.

10       “Communication media” typically embodies computer readable  
11 instructions, data structures, program modules, or other data in a modulated data  
12 signal, such as carrier wave or other transport mechanism. Communication media  
13 also includes any information delivery media. The term “modulated data signal”  
14 means a signal that has one or more of its characteristics set or changed in such a  
15 manner as to encode information in the signal. By way of example, and not  
16 limitation, communication media includes wired media such as a wired network or  
17 direct-wired connection, and wireless media such as acoustic, RF, infrared, and  
18 other wireless media. Combinations of any of the above are also included within  
19 the scope of computer readable media.

20       Although the description above uses language that is specific to structural  
21 features and/or methodological acts, it is to be understood that the invention  
22 defined in the appended claims is not limited to the specific features or acts  
23 described. Rather, the specific features and acts are disclosed as exemplary forms  
24 of implementing the invention.  
25